one or more of a look-up table (LUT) and/or registers of a CLB slice of the FPGA, i.e., utilizing the atomic elements of an FPGA.

Please replace the paragraph beginning on page 10, line 23 with the following:

According to one implementation, summing module generator performs maximal segmentation (virtual grouping of bits denoted by dashed lines 303) within the column to group bits in groups of 3, 2, or 1 bit(s), respectively. Three-bit groups are passed to a full adder for processing, while two-bit groups are passed to a half-adder for processing. Single bit columns are passed directly to an available register 312 within a CLB. In accordance with one aspect of the invention, summing module generator 222 utilizes standard routing analysis tools to identify the optimal atomic layout of each of the allocated elements 308-312 of the hybrid Wallace tree 306. According to one implementation, summing module generator is designed to minimize waste of atomic resources and allocates elements 308-312 in this regard. According to one implementation, summing module generator 222 prioritizes performance speed over waste and, as a result, seeks to minimize routing among and between atomic elements 206-212 implementing the hybrid summing module 306, even at the expense of some waste of atomic resources. In another implementation, resource conservation and performance are equally weighted, with resources allocated accordingly.

Please replace the equation beginning on page 18, line 7 with the following:

(2) $I_2=(a_2*c_2)-(b_2*d_2)$ and added to I_1; performed simultaneously with $Q_2=(a_2*d_2)+(b_2*c_2)$ and added to Q_1.

IN THE CLAIMS

Please cancel claim 1.

Please add the following new claims:

2. (New) A method comprising:

analyzing input terms on a bit-wise basis;

selecting resources to generate the summing module based, at least in part, on the analysis; and

designing a hyperpipelined series of Boolean function generators to implement a Wallace-architecture of full-adders, half-adders, and associated registers in the selected

resources, the series of Boolean function generators to combine the input terms to produce intermediate summation results.

3. (New) The method of claim 2 wherein analyzing input terms on a bit-wise basis comprises performing an analysis of bits of the input terms within each level of bit-significance.

4. (New) The method of claim 2 wherein selecting resources to generate the summing module comprises selecting atomic elements of a dedicated logic device in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

5. (New) The method of claim 4 wherein selecting atomic elements of a dedicated logic device in which to implement the Wallace-architecture comprises selecting atomic elements of a field programmable gate array (FPGA) in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

6. (New) The method of claim 4 wherein selecting atomic elements of a dedicated logic device in which to implement the Wallace-architecture comprises control logic in a device with a block of dedicated logic selecting atomic elements of the dedicated logic in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

7. (New) The method of claim 4 wherein selecting resources to generate the summing module based, at least in part, on the analysis further comprises:

determining a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented; and

selecting atomic elements of the dedicated logic device so as to reduce a number of atomic elements left unused in the design, to implement the Wallace architecture with a number of full-adders, half-adders, and associated registers that approaches the minimum number.

8. (New) The method of claim 4 wherein selecting resources to generate the summing module based, at least in part, on the analysis further comprises:

selecting atomic elements of the dedicated logic device to implement the Wallace-architecture of full-adders, half-adders, and associated registers; and

wherein designing the hyperpipelined series of Boolean function generators comprises assigning proximate atomic elements functions so as to reduce routing distances between stages of the Wallace-architecture.

9. (New) The method of claim 4 wherein selecting resources to generate the summing module based, at least in part, on the analysis further comprises:

determining a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented; and

wherein designing the hyperpipelined series of Boolean function generators comprises selecting atomic elements of the dedicated logic device to implement the Wallace-architecture with the minimal number of full-adders, half-adders, and associated registers while concurrently assigning proximate atomic elements to functions that result in reducing routing distances between stages of the Wallace-architecture.

10. (New) The method of claim 2 wherein selecting resources to generate the summing module comprises a system controller selecting one or more field programmable gate arrays (FPGAs) in the system in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers, and wherein designing the hyperpipelined series of Boolean function generators to implement the Wallace-architecture of full-adders, half-adders, and associated registers in the selected resources comprises the system controller designing the

hyperpipelined series of Boolean function generators to be implemented with the atomic elements of the selected FPGA(s).

11. (New) The method of claim 2 wherein designing the hyperpipelined series of Boolean function generators to implement the Wallace-architecture comprises designing the hyperpipelined series of Boolean function generators to increase grouping of bits of a same level of bit-significance of the input terms.

12. (New) The method of claim 2 wherein designing the hyperpipelined series of Boolean function generators comprises dynamically designing the hyperpipelined series of Boolean function generators to implement desired instances of the Wallace-architecture.

13. (New) The method of claim 2 further comprising:

implementing the design in the dedicated logic device by assigning the atomic elements of the dedicated logic device according to the design.

14. (New) The method of claim 2 further comprising:

utilizing the summing module with other components of a processing system;

identifying features of the other components that can be integrated into the design of the Wallace-architecture; and

wherein designing the hyperpipelined series of Boolean function generators includes integrating the identified features into the Wallace-architecture.

15. (New) The method of claim 14 wherein designing the hyperpipelined series of Boolean function generators includes adding accumulator bits from the other components to a summation result achieved by the Wallace-architecture.

16. (New) The method of claim 14 wherein designing the hyperpipelined series of Boolean function generators includes coupling the Wallace-architecture to output a summation result to at least one of the other components.

17. (New) An article of manufacture comprising a machine-accessible medium having content to provide instructions for generating a complex arithmetic summing module, the content to provide the instructions to cause an electronic system to:

analyze input terms on a bit-wise basis;

select resources to generate the summing module based, at least in part, on the analysis; and

design a hyperpipelined series of Boolean function generators to implement a Wallace-architecture of full-adders, half-adders, and associated registers in the selected resources, the series of Boolean function generators to combine the input terms to produce intermediate summation results.

18. (New) The article of manufacture of claim 17 wherein the content to provide instructions to cause the electronic system to analyze input terms on a bit-wise basis comprises the content to provide instructions to cause the electronic system to perform an analysis of bits of the input terms within each level of bit-significance.

19. (New) The article of manufacture of claim 17 wherein the content to provide instructions to cause the electronic system to select resources to generate the summing module comprises the content to provide instructions to cause the electronic system to select atomic elements of a dedicated logic device in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

20. (New) The article of manufacture of claim 19 wherein the content to provide instructions to cause the electronic system to select atomic elements of a dedicated logic device in which to implement the Wallace-architecture comprises the content to provide instructions to cause the electronic system to select atomic elements of a field programmable gate array (FPGA) in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

21. (New) The article of manufacture of claim 19 wherein the content to provide instructions to cause the electronic system to select atomic elements of a dedicated logic device in which to implement the Wallace-architecture comprises the content to provide instructions to cause the electronic system to direct control logic in a device with a dedicated logic array to select atomic elements of the dedicated logic in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

22. (New) The article of manufacture of claim 19 wherein the content to provide instructions to cause the electronic system to select resources to generate the summing module based, at least in part, on the analysis further comprises the content to provide instructions to cause the electronic system to:

determine a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented; and

select atomic elements of the dedicated logic device so as to reduce a number of atomic elements left unused in the design, to implement the Wallace architecture with a number of full-adders, half-adders, and associated registers that approaches the minimum number.

23. (New) The article of manufacture of claim 19 wherein the content to provide instructions to cause the electronic system to select resources to generate the summing module

based, at least in part, on the analysis further comprises the content to provide instructions to cause the electronic system to:

select atomic elements of the dedicated logic device to implement the Wallace-architecture of full-adders, half-adders, and associated registers; and

wherein the content to provide instructions to cause the electronic system to design the hyperpipelined series of Boolean function generators comprises the content to provide instructions to cause the electronic system to assign proximate atomic elements functions so as to reduce routing distances between stages of the Wallace-architecture.

24. (New) The article of manufacture of claim 19 wherein the content to provide instructions to cause the electronic system to select resources to generate the summing module based, at least in part, on the analysis further comprises the content to provide instructions to cause the electronic system to:

determine a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented;

wherein the content to provide instructions to cause the electronic system to design the hyperpipelined series of Boolean function generators comprises the content to provide instructions to cause the electronic system to select atomic elements of the dedicated logic device to implement the Wallace-architecture with the minimal number of full-adders, half-adders, and associated registers and concurrently to assign proximate atomic elements to functions to result in reducing routing distances between stages of the Wallace-architecture.

25. (New) The article of manufacture of claim 17 wherein the content to provide instructions to cause the electronic system to select resources to generate the summing module comprises the content to provide instructions to cause the electronic system to direct a controller

to select one or more field programmable gate arrays (FPGAs) coupled with the controller in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers, and wherein the content to provide instructions to cause the electronic system to design the hyperpipelined series of Boolean function generators to implement the Wallace-architecture of full-adders, half-adders, and associated registers in the selected resources comprises content to provide instructions to cause the electronic system to direct the controller to design the hyperpipelined series of Boolean function generators to be implemented with the atomic elements of the selected FPGA(s).

26. (New) The article of manufacture of claim 17 wherein the content to provide instructions to cause the electronic system to design the hyperpipelined series of Boolean function generators to implement the Wallace-architecture comprises the content to provide instructions to cause the electronic system to design the hyperpipelined series of Boolean function generators to increase grouping of bits of a same level of bit-significance of the input terms.

27. (New) The article of manufacture of claim 17 wherein the content to provide instructions to cause the electronic system to design the hyperpipelined series of Boolean function generators comprises the content to provide instructions to cause the electronic system to dynamically design the hyperpipelined series of Boolean function generators to implement desired instances of the Wallace-architecture.

28. (New) The article of manufacture of claim 17 further comprising the content to provide instructions to cause the electronic system to:

implement the design in the dedicated logic device by assigning the atomic elements of the dedicated logic device according to the design.